Governance Participation Analysis

1. **Goals:**
    a. Increase Participation Rates in On-Chain Voting
    b. Enhance DAO Legitimacy
    c. To understand voting patterns and contributions of delegates to on chain voting.

2. **Milestones:**

    a. Data collection :
        i. a. Gathering what kind of dataset is required and useful for the analysis.
        ii. b. Finding a reliable source for the required data.
        iii. c. Api calling and python code.
        iv. d. Adding the paginations and limits for the dataset
        v. e. Data trimming and cleaning
        vi. f. Verifying the data collected to be accurate and proper.

    b. Data analysis:
        i. Creating a basics data insights for the gathered data sets and tables
        ii. Conduct comprehensive analytics on on-chain voting participation rates.
        iii. Examine correlations between variables related to participation patterns to provide recommendations and insights.
        iv. Analyse how Tally affects participation in on-chain voting, which categories of users take part more often, and who publishes proposals most often.

DATA GATHERING:

1. Gathering what kind of dataset is required and useful for the analysis

Our purpose is to gather the data about the participation rate on the on chain tally platform.

Kind to dataset required:

a. Voter data who are voting on the on-chain tally platform
   i. Voting information like voting weight given to particular proposal
   ii. Clear mapping of proposals and voter
b. Delegators and delegates data
   i. A detailed information of voting power change over the period of time
   ii. Delegators count
c. Unique voters:
   i. Delegators associated with particular voter id (if no delegators then voter will be considered as ex delegator)
   ii. Total votes given(for ,against ,abstain)
d. Proposals data
   i. All the proposals created
   ii. Voters associated with proposals
   iii. Vote counts and proposers information
e. Token holder's data:
   i. Total erc20 token holders which are currently active.
   ii. Balance amount and their wallet account address

2. Finding a reliable source for the required data.

a. Tally on-chain platform provide its own Api to fetch proposal voters and delegates data from its
b. For the token holders data bitquery api is used to fetch the data:

3. Api calling and python code.

a. Tally api information :
   i. Api endpoint : 'https://api.tally.xyz/query'
   ii. headers : {Api-key: ""}
   iii. query: [graphql_query]
   iv. variables
   v. payload :{ api_url , json={'query': query, 'variables': variable}, headers=header}

b. Bitquery api information :
    i. API-endpoint : "streaming.bitquery.io"
    ii. Header : headers = {
    iii. 'Content-Type': '', 'X-API-KEY': '','Authorization': ' ' }
    iv. payload = { “query": ' ', 'variable ': '{} '}

4. Adding the paginations and limits for the dataset

a. Json data loaded from the api call were incomplete after tallying from the front end
b. In the json response the data are coming into the chunks of pages
c. The huge amount of data is not able to be loaded at once into the json response file
d. Solution:
e. To overcome this problem we need to load the json data to the csv file .
f. Query each json file in the loop and keep adding the data to the csv file.
g. The following tables and there pagination parameters:

Table	Limit
Proposal	10
Voters	10000
Delegates	20
Token holders	20000

h. A single Api call made takes around 1 sec of time.

5. Data trimming and cleaning :

a. The raw data fetched from the direct api call my included :
    i. Duplicates
    ii. Missing entries
    iii. Invalid Data types
    iv. Difference in Character cases
b. To overcome this a python script for data cleaning is used
c. Other features of python script:
    i. Add/removal for columns
    ii. Sorting
    iii. Merging tables
    iv. Modifying columns

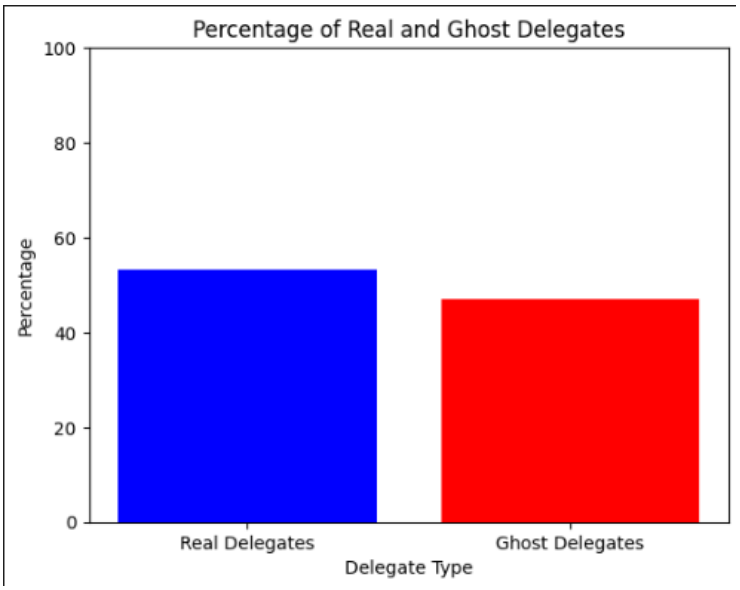
6. Verifying the data collected to be accurate and proper.

a. Compare the collected data with other independent sources to ensure consistency, confirming that the information aligns across different platforms or records.
b. Randomly select subsets of the data and validate its quality through manual inspection or automated checks, providing an overview of the data's reliability.
c. Visualise the data using charts or graphs, enabling easy identification of any trends, patterns, or inconsistencies that may require further investigation.
d. Conduct statistical analysis to assess the data's reliability and accuracy, using mathematical techniques to validate its consistency with expected distributions or properties.
e. **Issues faced regarding data collection:**
    i. Incomplete data
    ii. Wrong formatting
    iii. Significant amount of time taken to fetch all data
    iv. Incompetency with original data information provided on front end.

DATA ANALYSIS:

1. Ghost delegates :

The delegates who have received the voting power but haven't voted are considered as a ghost delegate.
About half of the delegates having the voting power are not participating in the voting process at.



Total delegates :158.73k

Ghost delegates : 74.67k

Delegates who have voted 84.65k (Real delegates)

Here the telly data have 46.88% ghost delegates out of total delegates

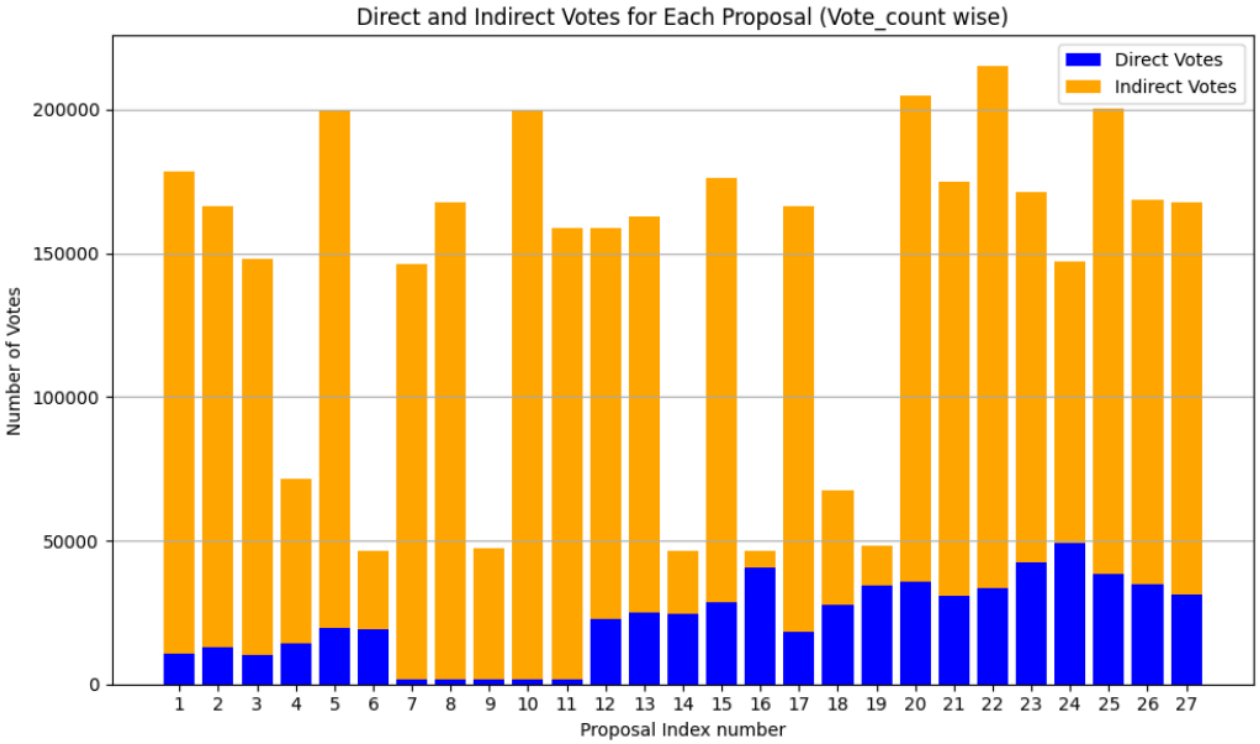
2. Voters having the maximum impact on the voting process.

Identify the top voters based on their voting weight.

	voter.address	voter.name	weight
1447	0x0eb5b03c0303f2f47cd81d7be4275af8ed347576	Treasure	4.799599e+08
23885	0xf4b0556b9b6f53e00a1fdd2b0478ce841991d8fa	olimpio.eth	3.977026e+08
2745	0x1b686ee8e31c5959d9f5bbd8122a58682788eead	L2BEAT 💗	3.075374e+08
12832	0x839395e20bbb182fa440d08f850e6c7a8f6f0780	Griff Green 🙏🍸🎗️💜	2.163766e+08
18318	0xbbe98d590d7eb99f4a236587f2441826396053d3	PlutusDAO	2.026802e+08
17730	0xb5b069370ef24bc67f114e185d185063ce3479f8	Frisson	1.714908e+08
4633	0x2ef27b114917dd53f8633440a7c0328fef132e2f	MUX Protocol	1.536926e+08
16174	0xa5df0cf3f95c6cd97d998b9d990a86864095d9b0	Blockworks Research	1.406700e+08
2509	0x190473b3071946df65306989972706a4c006a561	ChainLinkGod	1.151511e+08
20620	0xd333bc5c9670c9ceb18f9a2cf02c6e86807a8227	MaxLomu	7.645340e+07

3. **Direct voters (delegates who have given the vote) and indirect voters(delegators giving their voting power to delegate) for a particular proposal.**

Direct voters : delegates who are voting on the proposal.
Indirect voters :delegators trusting the delegates to give them voting power and cast a vote on behalf of them.



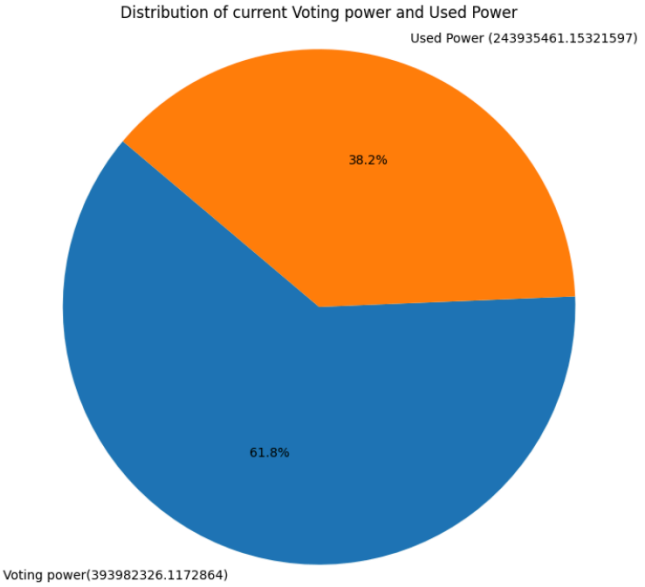
Total active token holder : 918k
Delegates : 157k

Note: Over 82% of the token holders are delegating their voting power to someone else.
According to the dataset the indirect voters are much more.

4. Voters participation in the voting process (based on the voting power they have and how much they have used).

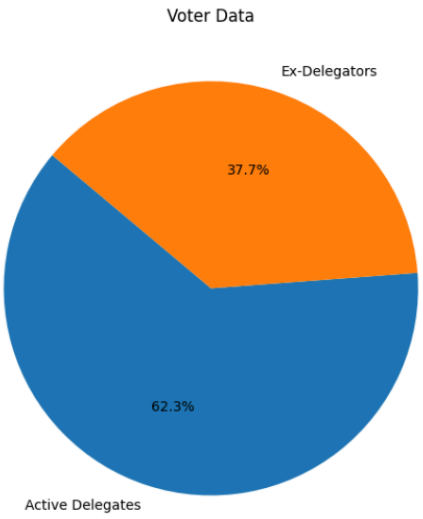
In the DAO's voting process ,the voter participation is just about 40%.

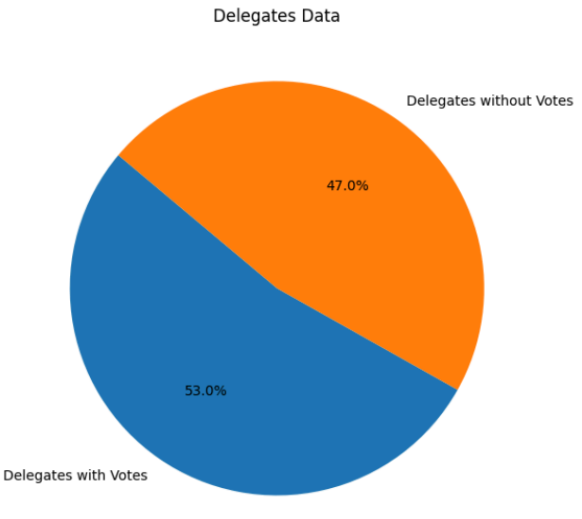
Voter's participation is calculated as the percentage of the sum of voting power whoever have voted before to the total voting power delegated to the voters.



5. List of the active and ex-delegates who had voted previously but delegated.

There are about 38 % of ex-delegates who had voted before and by receiving the voting power but they are not active.
Voter's participation is calculated as the percentage of the sum of voting power whoever has voted before to the total voting power delegated to the voters.





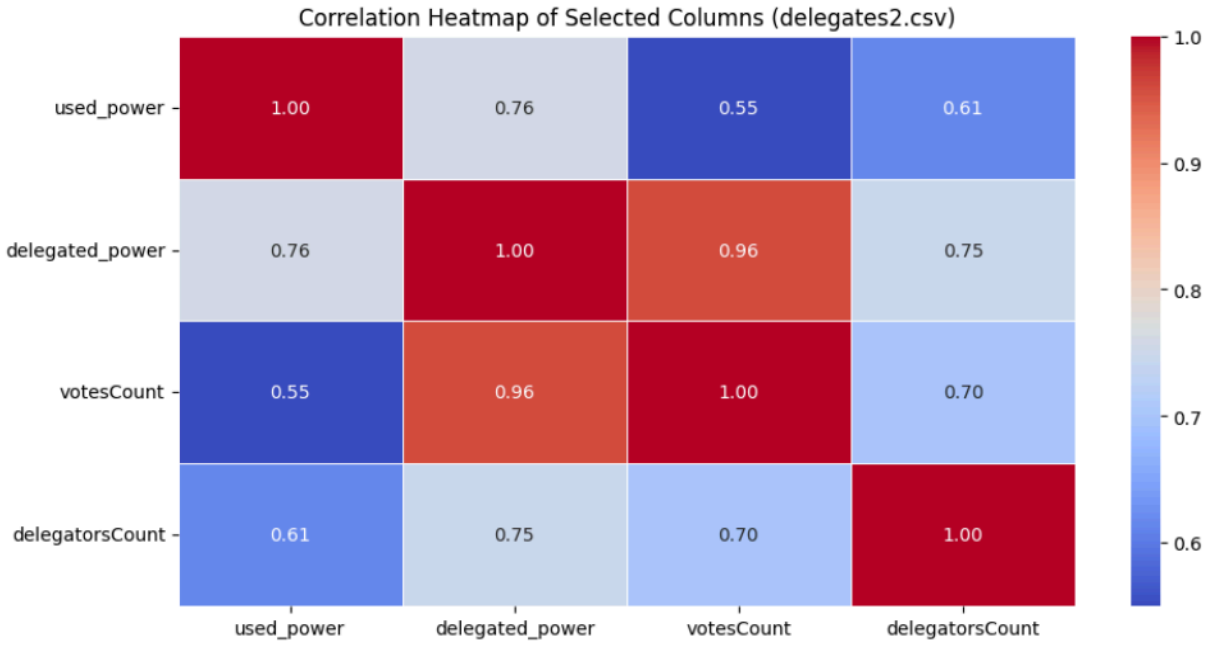
Visuals: The left chart shows the voter's data while the right chart shows delegates data

Active delegates count	:84k	Ex-delegates count	:51k
Ghost-delegates count	:74k	Total_voters	:135k

6. **Used_power / total delegated power / votescount(voting power) analysis for delegates (with correlation table).**

used_power : total power used by the delegates up to now for voting or delegating votes to someone else
delegated power : total power which is delegates to the user up to now
votesCount : how much voting power doe the voter is holding currently

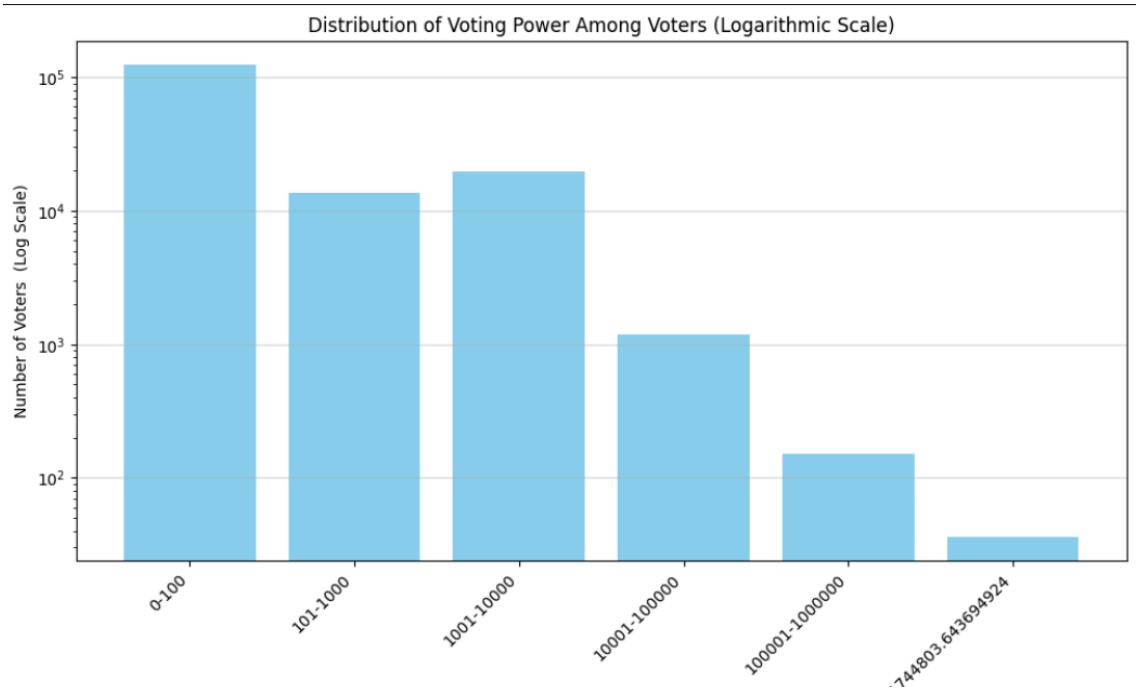
Correlation table:



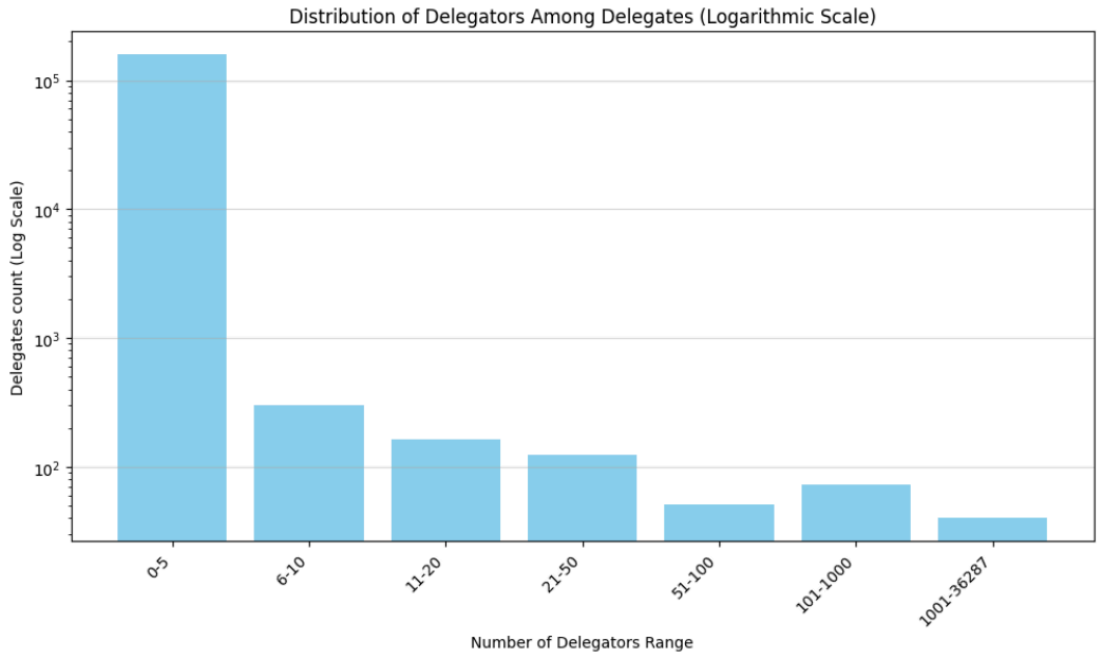
Here the voting power are updated at first day of every month so the record of the transaction happened inside on the internal days are not considered

7. Distribution of voting power/delegators among the voters:

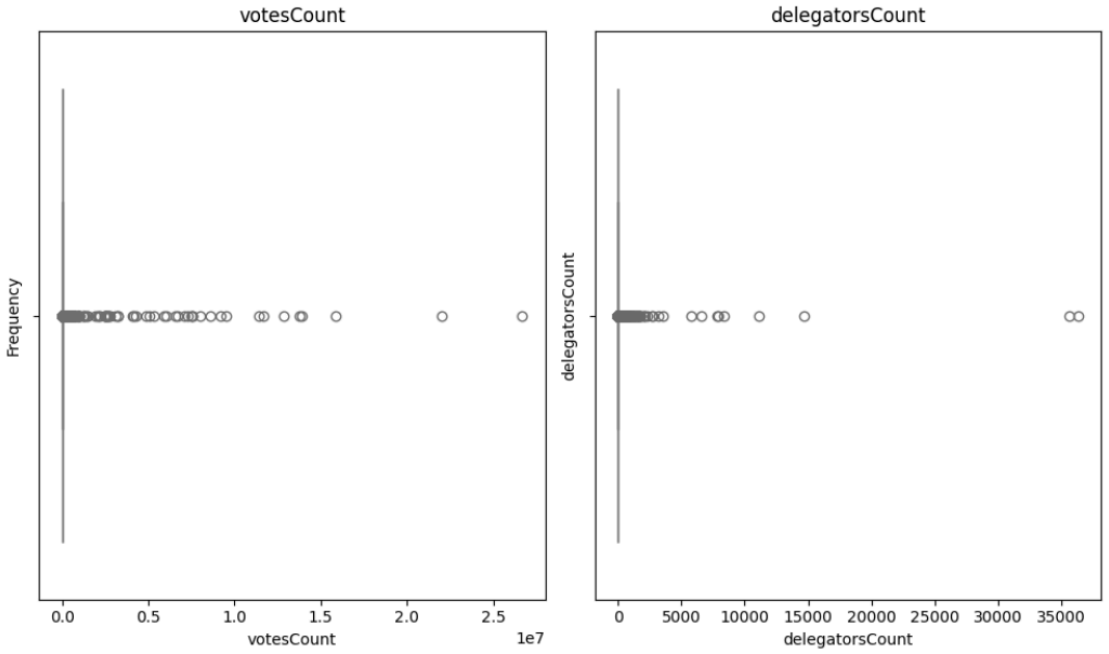
Observation: Here we observe that a few percentage of people are having high voting power and maximum number of people are having the voting power below 10.



Similarly for the delegators count possessed by each of the voters is also observed to show a similar trend



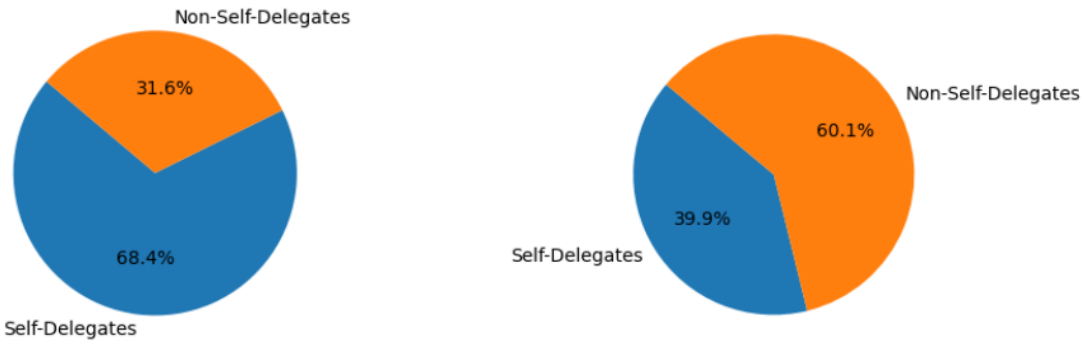
Box-plot:



Statistical Summary:

	used_power	delegated_power	votesCount	delegatorsCount
count	1.587380e+05	1.587380e+05	1.587380e+05	158738.000000
mean	1.536717e+03	4.018684e+03	2.481966e+03	2.469812
std	6.103667e+04	1.845017e+05	1.437739e+05	144.602291
min	0.000000e+00	1.000000e-18	0.000000e+00	1.000000
25%	0.000000e+00	1.213236e+00	1.004149e+00	1.000000
50%	0.000000e+00	9.572819e+00	3.000000e+00	1.000000
75%	2.600960e+01	6.258523e+02	2.691822e+01	1.000000
max	1.174480e+07	3.212906e+07	2.666241e+07	36287.000000

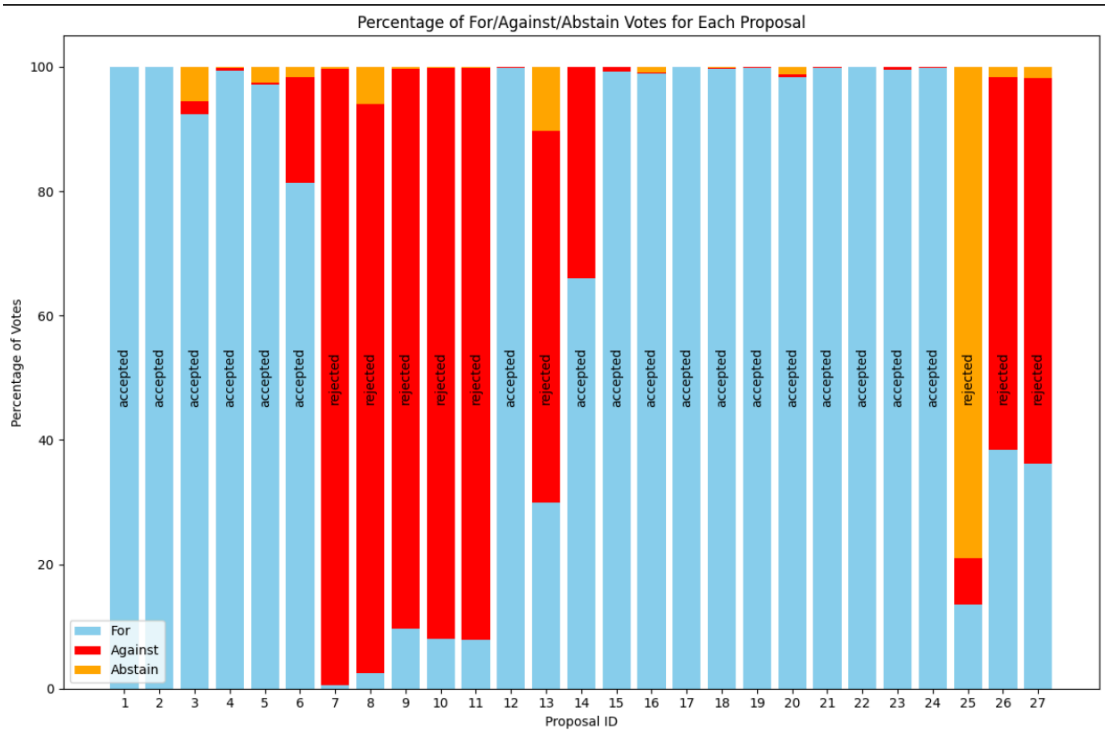
Self delegates and proper delegates:



Based on the voting power **Based on delegatorsCount**

8. Proposal_data general analysis :

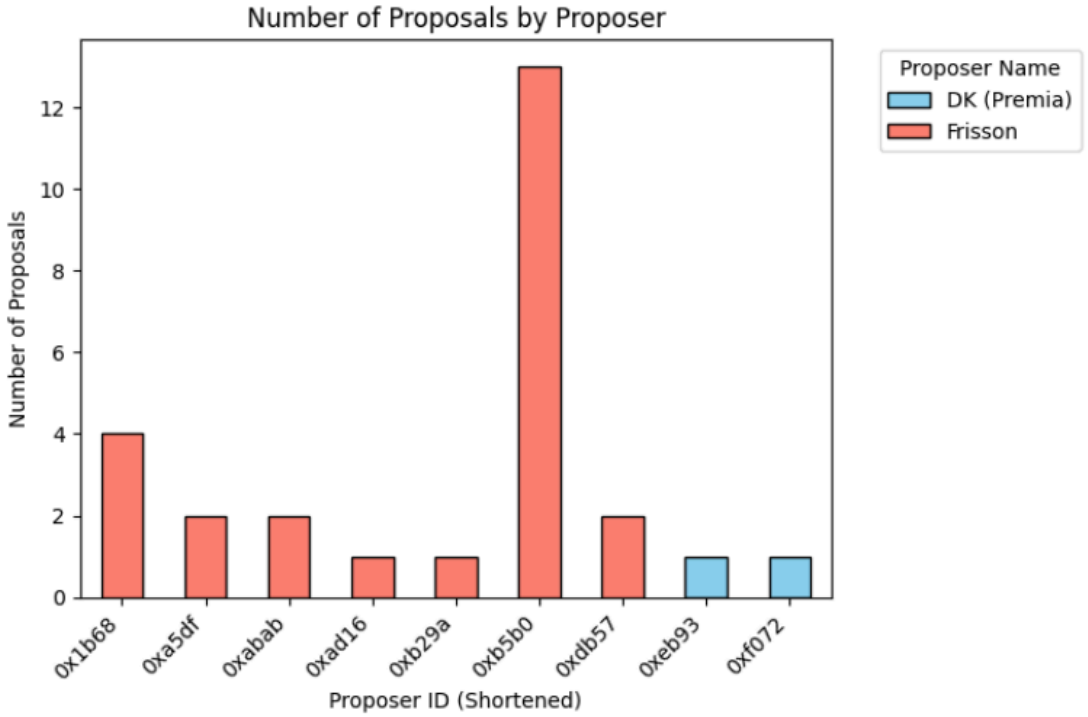
a. For / against / abstain votes:



Every proposal should qualify the qurum's requirement suppose for the proposal 25th is rejected even though the FOR count is more then AGAINST votes

81.1M out of 81.47M votes are given which didn't fulfils the requirement

b. Categories of users creating the proposals most often:



Conclusion :

The overall voter's participation rate on the tally on-chain platform is about 38% of total. The ghost delegates are the major reason for the lack of participation rate these ghost delegates Comprises 46% of total delegates.
The indirect voters which are the delegators who delegated this power to the delegates. About 27% of the voting power are possessed by these ghost delegates, leading to this problem. If these delegates show interest and participate in the voting process we could increase the voting participation to 45%-50%.

Resources:

Data sets :

- Tally API docs : https://apidocs.tally.xyz/ ,
- API endpoint : https://api.tally.xyz/query
- Bitquery API endpoint: https://streaming.bitquery.io/graphql
- Github link for dataset and codes:
- Matplotlib python library : https://matplotlib.org/
- Arbiscan : https://arbiscan.io/
- Tally platform : https://www.tally.xyz/